Karachi Institute of Economics & Technology
College of Computing & Information Sciences

Numerical Computing & Analysis

Assignment

Class ID :	**Student ID : 7791**
Student Name : Muhammad Umair	

```
import math
import matplotlib. pyplot as plt
%matplotlib inline
import numpy as np
```

Q1 : Solve the following and find the value of 's' using python and math library. 1 Marks

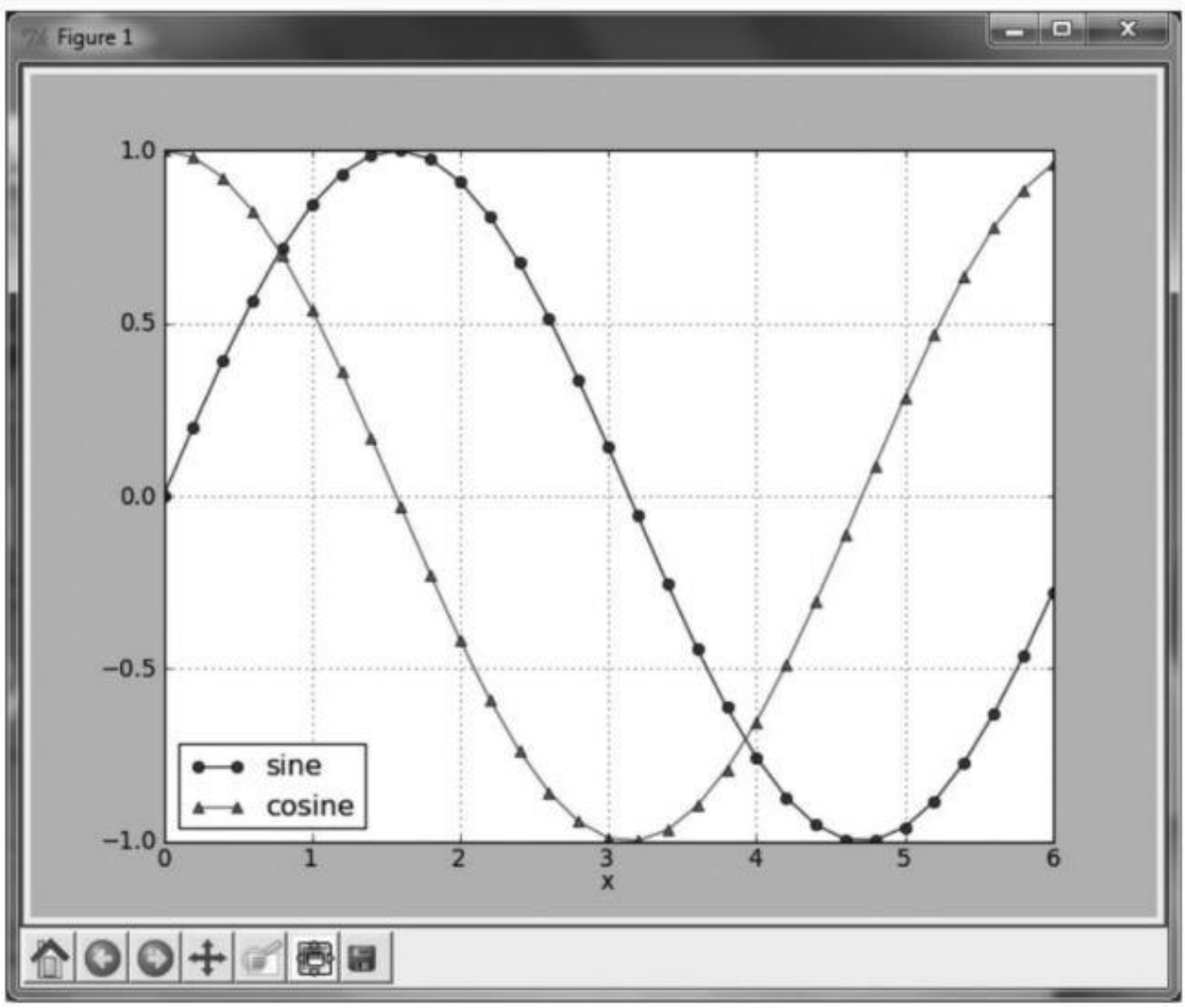
$$s = \sum_{i=0}^{100} \sqrt{\frac{i\pi}{100}} \sin \frac{i\pi}{100}$$

```
S = 0
For I in range(0,100);
        Sq=math.sqrt(i*math.pi/100)
        Sin=math.sin((i*math.pi)/100)
        S += s+(sq*sin)
Print (s)
```

1 .1587596243801539e+28

Q2 : Plot the graph below using python & matplotlib package. 1 Marks

```
X = np.linspace(0,2* np.pi,100)
Y = np.sin(x)
Y1 = np.cos(x)
Plt.plot(x,y,'<',color='red',label='sine')
Plt.plot(x,y1,'o',color=blue,label='cosine')
Plt.xlabel('x')
Plt.grid(True)
Plt.legend(loc="best")
Plt.show()
```



Q3: As shown in figure, find the lower and upper matrix using LU Decomposition of Marix 'A' with python. 3 Marks

```
MAX = 100
Def luDecomposition(mat,n):
        Lower = [[0 for x in range(n)]for y in range(n)];
        upper = [[0 for x in range(n)]for y in range(n)];
        for row in range(n):
            for col in range(row,n):
                sum = 0
                for j in range(row):
                    sum += (lower[row][j] * upper[j][col]);
                upper[row][col] = mat[row][col] – sum;
            for col in range(row,n):
                if(row == col):
                    lower[row][row] = 1;
                else:
                    sum = 0
                    for j in range(row):
                        sum +=(lower[col][j] * upper[j][row]);
                    lower[col][row]=int((mat[col][row] – sum)/2)
        print("lower")
        for I in range(n):
            for j in range(n):
                print(lower[i][j],end = "\t")
            print(" ")
        print("upper")
        for I in range(n):
            for j in range(n):
                print(upper[i][j],end="\t")
            print(" ")
mat = [[4,-1,0],[-1,4,-1],[0,-1,4]];
liDecomposition(mat,3);
```

lower		
1	0	0
0	1	0
0	0	1
Upper		
4	-1	0
0	4	-1
0	0	4

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

Figure

$$\mathbf{A} = \mathbf{LU} = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

Q4: Solve the equation 3 Marks

$$\begin{bmatrix} 4 & -1 & 1 \\ -1 & 4 & -2 \\ 1 & -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ -1 \\ 5 \end{bmatrix}$$

Using Guass-Seidel Method.

```
def seidel(a, x ,b) :
    n= len(a)
    for j in range(0, n) :
        d = b[j]
        for I in range(0, n ):
                if(j != i):
                    d -=a[j][i]*x[i]
        x[j] = d/a[j][j]
    return x
n = 3
a = []
b = []
x= [0,0,0]
a=[[4,-1,1],[-1,4,-2],[1,-2,4]]
b=[12,-1,5]
for i in range(0, 25):
        x=seidel(a, x, b)

        print(x)
```

```
[3.0, 0.5, 0.75]
[2.9375, 0.859375, 0.9453125]
[2.978515625, 0.96728515625, 0.989013671875]
[2.99456787109375, 0.9931488037109375, 0.9979324340820312]
[2.9988040924072266, 0.9986672401428223, 0.9996325969696045]
[2.9997586607933044, 0.9997559636831284, 0.9999383166432381]
[2.9999544117599726, 0.9999577612616122, 0.999990277690813]
[2.9999918708927, 0.9999931065685814, 0.9999985855611158]
[2.9999986302518664, 0.9999989503435245, 0.9999998176087956]
[2.999999783183682, 0.9999998546003184, 0.9999999815042386]
[2.99999996827402, 0.9999999828206243, 0.9999999993418072]
[2.9999999958697043, 0.9999999986383297, 1.0000000003517386]
[2.9999999999571648, 1.0000000000687814, 1.0000000001414788]
[2.999999999981825, 1.0000000000661957, 1.0000000000376414]
[3.0000000000071387, 1.0000000000206053, 1.000000000008518]
[3.000000000003022, 1.0000000000050147, 1.000000000001752]
[3.000000000000816, 1.0000000000000108, 1.0000000000003362]
[3.000000000000186, 1.0000000000002145, 1.0000000000000608]
[3.0000000000000386, 1.0000000000000004, 1.0000000000000102]
[3.000000000000007, 1.0000000000000069, 1.0000000000000018]
[3.0000000000000013, 1.0000000000000013, 1.0000000000000004]
[3.0000000000000004, 1.0000000000000004, 1.0]
[3.0, 1.0, 1.0]
[3.0, 1.0, 1.0]
[3.0, 1.0, 1.0]
```

Q5: Find integration using Simpson 1/3 Rule. 2 Marks

$$\int_a^b f(x)dx \approx \frac{(b-a)}{6}\left(f(a) + 4f\frac{(a+b)}{2} + f(b)\right)$$

a = 8.4, b=9.6. n=6. f(x) = log x

```
def func( x ):
    return math . log(x)
def simpsons_( ll, ul, n ) :
        h=(ul – ll)/n
        x=list()
        fx=list()
        i=0
      while i<=n:
        x. append(ll + I * h)
        fx. append (func (x[i]))
        I += 1
      Res = 0
      I = 0
      While i<= n:
        If I == 0 or I  == n:
                Res += fx[i]
        Elseif I % 2 != 0:
                Res += 4 * fx[i]
        Else:
                Res += 2 * fx[i]
        I+= 1
      Res = res * (h / 3)
      Return res

    A = 8.4
    B = 9.6
    N = 6
    print(simpsons_(a, b, n))
    # its write 6 digit before point
    print("%.6f"% simpsons_(a, b, n) )
```

```
2.6357794063258604
2.635779
```